IN THE CLAIMS

Please cancel Claims 2, 12, 14, and 30-32 without prejudice, amend Claims 1, 13, and 33, and add new Claims 40-42 as follows:

1. (Currently amended) A method of transmitting data across a high-speed serial bus, the method comprising:

in accordance with a first TX symbol clock:

generating a 10-bit symbol on an IEEE 1394-compliant PHY having a port interface;

placing the generated 10-bit symbol on the port interface;

scrambling the 10-bit symbol;

encoding the 10-bit symbol;

placing the 10-bit symbol in a FIFO;

in accordance with a second TX clock, the second TX clock running at a different speed than the first TX clock:

allowing the 10-bit symbol to be removed from the FIFO only on four out of every five TX clock cycles associated with the second clock;

removing the 10-bit symbol from the FIFO;

deriving an 8-bit byte from the removed 10-bit symbol; and

sending the 8-bit byte to an IEEE 802.3-compliant PHY.

2. (Cancelled)

3. (Original) The method of claim 1, wherein a null 10-bit symbol is placed in the FIFO if there are no 10-bit symbols present in the FIFO.

4. (Original) The method of claim 1, wherein the 8-bit byte is derived from the 10-bit symbol by using 8 bits from the extracted 10-bit symbol, and the two remaining bits are stored.

5. (Original) The method of claim 4, wherein a second 8-bit byte is derived by extracting from the FIFO a second 10-bit symbol and assembling an 8-bit byte from the stored two bits and six bits from the extracted second 10-bit symbol; the four remaining bits from the extracted second symbol are stored; and the second 8-bit byte is sent to the IEEE 802.3-compliant PHY.

6. (Original) The method of claim 5, wherein a third 8-bit byte is derived by extracting from the FIFO a third 10-bit symbol and assembling an 8-bit byte from the four stored bits and

four bits from the third extracted symbol; the six remaining bits from the extracted third symbol are stored; and the third 8-bit byte is sent to the IEEE 802.3-compliant PHY.

7. (Original) The method of claim 6, wherein a fourth 8-bit byte is derived by extracting from the FIFO a fourth 10-bit symbol, and assembling an 8-bit byte from the six stored bits and 2 bits from the extracted fourth 10-bit symbol; the eight remaining bits from the extracted fourth symbol are stored; and the fourth 8-bit byte is sent to the IEEE 802.3-compliant PHY.

8. (Original) The method of claim 7, wherein a fifth 8-bit byte is derived from the stored eight remaining bits and sent to the IEEE 802.3-compliant PHY.

9. (Original) The method of claim 1, further comprising, in accordance with a phase amplitude modulation clock, sending the received 8-bit byte from the IEEE 802.3-compliant PHY to a device in accordance with a phase amplitude modulation clock.

10. (Previously presented) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving an 8-bit byte on an 802.3-compliant PHY;

in accordance with a GMII RX clock:

if the received 8-bit byte contains a null symbol, then deleting the null symbol;

else if the received 8-bit byte does not contain a null symbol, then storing the 8-bit byte in a first register;

receiving a second 8-bit byte that does not contain a null symbol and storing the second 8-bit byte in a second register;

assembling a 10-bit symbol from the 8-bit byte stored in the first register and appending two bits from the 8-bit byte stored in the second register; and

placing the assembled 10-bit symbol in a first FIFO;

in accordance with a second clock:

removing the 10-bit symbol from the first FIFO;

performing 8B10B and control decoding on the removed 10-bit symbol; and

placing the decoded 10-bit symbol in a second FIFO;

in accordance with a third clock:

removing the decoded 10-bit symbol from the second FIFO; and

sending the decoded 10-bit symbol to an IEEE 1394-compliant PHY.

11. (Original) The method of claim 10, wherein the second clock is phase locked to the third clock.

12. (Cancelled)

13. (Currently amended) A method of transmitting data across a high-speed serial bus, the method comprising:

in accordance with a first TX symbol clock:

generating a 10-bit symbol on an IEEE 1394-compliant PHY having a port interface;

placing the generated 10-bit symbol on the port interface;

performing flagged encoding the 10-bit symbol;

placing the 10-bit symbol in a FIFO;

in accordance with a second TX clock, the second TX clock running at a different speed than the first TX clock:

allowing the 10-bit symbol to be removed from the FIFO only on four out of every five TX clock cycles associated with the second clock;

removing the 10-bit symbol from the FIFO;

deriving an 8-bit byte from the removed 10-bit symbol; and

sending the 8-bit byte to an IEEE 802.3-compliant PHY.

14. (Cancelled)

15. (Original) The method of claim 13, wherein a null 10-bit symbol is placed in the FIFO if there are no 10-bit symbols present in the FIFO.

16. (Original) The method of claim 13, wherein the 8-bit byte is derived from the 10-bit symbol by using 8 bits from the extracted 10-bit symbol, and the two remaining bits are stored.

17. (Original) The method of claim 16, wherein a second 8-bit byte is derived by extracting from the FIFO a second 10-bit symbol and assembling an 8-bit byte from the stored two bits and six bits from the extracted second 10-bit symbol; the four remaining bits from the extracted second symbol are stored; and the second 8-bit byte is sent to the IEEE 802.3-compliant PHY.

18. (Original) The method of claim 17, wherein a third 8-bit byte is derived by extracting

from the FIFO a third 10-bit symbol and assembling an 8-bit byte from the four stored bits and four bits from the third extracted symbol; the six remaining bits from the extracted third symbol are stored; and the third 8-bit byte is sent to the IEEE 802.3-compliant PHY.

19. (Original) The method of claim 18, wherein a fourth 8-bit byte is derived by extracting from the FIFO a fourth 10-bit symbol, and assembling an 8-bit byte from the six stored bits and 2 bits from the extracted fourth 10-bit symbol; the eight remaining bits from the extracted fourth symbol are stored; and the fourth 8-bit byte is sent to the IEEE 802.3-compliant PHY.

20. (Original) The method of claim 19, wherein a fifth 8-bit byte is derived from the stored eight remaining bits and sent to the IEEE 802.3-compliant PHY.

21. (Original) The method of claim 13, further comprising, in accordance with a phase amplitude modulation clock, sending the received 8-bit byte from the IEEE 802.3-compliant PHY to a device in accordance with a phase amplitude modulation clock.

22. (Original) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving an 8-bit byte on an IEEE 802.3-compliant PHY;

in accordance with a GMII RX clock:

if the received 8-bit byte contains a null symbol, then deleting the null symbol;

else if the received 8-bit byte does not contain a null symbol, then storing the 8-bit byte in a first register;

receiving a second 8-bit byte that does not contain a null symbol and storing the second 8-bit byte in a second register;

assembling a 10-bit symbol from the 8-bit byte stored in the first register and appending two bits from the 8-bit byte stored in the second register;

performing flagged decoding on the assembled 10-bit symbol and placing the assembled 10-bit symbol in a FIFO;

in accordance with a second clock:

removing the 10-bit symbol from the first FIFO; and

sending the decoded 10-bit symbol to an IEEE 1394-compliant PHY.

23. (Original) The method of claim 22, wherein a received data valid state is asserted on the IEEE 802.3-compliant PHY.

24. (Original) The method of claim 22, wherein the FIFO compensates for ppm differences between the IEEE 802.3-compliant PHY and the IEEE 1394-compliant PHY.

25. – 28. (Canceled)

29. (Previously presented) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving an 8-bit byte on an 802.3-compliant PHY;

in accordance with a GMII RX clock:

if the received 8-bit byte contains a null symbol, then deleting the null symbol;

else if the received 8-bit byte does not contain a null symbol, then storing the 8-bit byte in a first register;

receiving a second 8-bit byte that does not contain a null symbol and storing the second 8-bit byte in a second register;

assembling a 10-bit symbol from the 8-bit byte stored in the first register and appending two bits from the 8-bit byte stored in the second register; and

placing the assembled 10-bit symbol in a first FIFO;

in accordance with a second clock:

removing the 10-bit symbol from the first FIFO;

performing 8B10B and control decoding on the removed 10-bit symbol; and

placing the decoded 10-bit symbol in a second FIFO;

in accordance with a third clock:

removing the decoded 10-bit symbol from the second FIFO; and

sending the decoded 10-bit symbol to an IEEE 1394-compliant PHY;

wherein the second clock is phase locked to the third clock, the frequency of null character deletion is used to control a phased locked loop, and the phase locked loop is associated with the second clock.

30. – 32. (Cancelled)

33. (Currently amended) A method of transmitting data across a high-speed serial bus, the method comprising:

in accordance with a first clock:

generating a first multi-bit symbol on first physical interface having a port only on fifty-eight out of every fifty-nine clock cycles associated with the first clock, wherein the multi-bit symbol is compliant with a first transmission protocol;

placing the generated first symbol on the port;

scrambling the first symbol;

encoding the scrambled first symbol;

placing the scrambled symbol in a buffer;

in accordance with a second clock running at a different speed than the first clock:

deriving a multi-bit byte from the scrambled symbol; and

sending the multi-bit byte to a second physical interface, the second interface utilizing a different communication protocol than the first interface.

34. (Previously presented) The method of claim 33, further comprising removing the scrambled symbol from the buffer before performing said act of deriving.

35. (Previously presented) The method of claim 33, further comprising placing a null multi-bit symbol in the buffer if there are no scrambled multi-bit symbols present in the buffer.

36. (Previously presented) The method of claim 33, wherein the multi-bit byte is derived from the first symbol by using 8 bits from the scrambled symbol.

37. (Previously presented) The method of claim 33, further comprising, in accordance with a phase amplitude modulation clock, sending the received multi-bit byte from the second interface to a device in accordance with a phase amplitude modulation clock.

38. (Previously presented) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving an 8-bit byte on an 802.3-compliant PHY;

in accordance with a GMII RX clock:

if the received 8-bit byte contains a null symbol, then deleting the null symbol;

else if the received 8-bit byte does not contain a null symbol, then storing the 8-bit byte in a first register;

receiving a second 8-bit byte that does not contain a null symbol and storing the second 8-bit byte in a second register;

assembling a 10-bit symbol from the 8-bit byte stored in the first register and appending two bits from the 8-bit byte stored in the second register; and

placing the assembled 10-bit symbol in a first FIFO;

in accordance with a second clock:

removing the 10-bit symbol from the first FIFO;

performing 8B10B and control decoding on the removed 10-bit symbol; and

placing the decoded 10-bit symbol in a second FIFO;

in accordance with a third clock:

removing the decoded 10-bit symbol from the second FIFO; and

sending the decoded 10-bit symbol to an IEEE 1394-compliant PHY;

wherein the second clock is phase locked to the third clock; and

wherein frequency of null character deletion is used to control a phased locked loop, the phase locked loop associated with the second clock.

39. (Previously presented) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving an 8-bit byte on an 802.3-compliant PHY;

in accordance with a clock:

if the received 8-bit byte contains a null symbol, then deleting the null symbol;

else storing the 8-bit byte in a first location if it does not contain a null symbol, then;

receiving a second 8-bit byte that does not contain a null symbol and storing the second 8-bit byte in a second location;

assembling a 10-bit symbol from the 8-bit byte stored in the first location and appending two bits from the 8-bit byte stored in the second location; and

placing the assembled 10-bit symbol in a first buffer;

in accordance with a second clock:

removing the 10-bit symbol from the first buffer;

processing the removed 10-bit symbol to accomplish decoding thereof; and

placing the decoded 10-bit symbol in a second buffer;

in accordance with a third clock:

removing the decoded 10-bit symbol from the second buffer; and

sending the decoded 10-bit symbol to an IEEE 1394-compliant PHY;

wherein the second clock comprises a predetermined relationship to the third clock; and

wherein frequency of null character deletion is used to control a phased locked loop, the phase locked loop associated with the second clock.

40. (New) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving an 8-bit byte on an 802.3-compliant PHY;

in accordance with a GMII RX clock:

if the received 8-bit byte contains a null symbol, then deleting the null symbol;

else if the received 8-bit byte does not contain a null symbol, then storing the 8-bit byte in a first register;

receiving a second 8-bit byte that does not contain a null symbol and storing the second 8-bit byte in a second register;

assembling a 10-bit symbol from the 8-bit byte stored in the first register and appending two bits from the 8-bit byte stored in the second register; and

placing the assembled 10-bit symbol in a first FIFO;

in accordance with a second clock:

removing the 10-bit symbol from the first FIFO;

performing 8B10B and control decoding on the removed 10-bit symbol; and

placing the decoded 10-bit symbol in a second FIFO;

in accordance with a third clock:

removing the decoded 10-bit symbol from the second FIFO; and

sending the decoded 10-bit symbol to an IEEE 1394-compliant PHY;

wherein the second clock is phase locked to the third clock; and wherein the frequency of null character deletion is used to control a phased lock loop, the phase locked loop associated with the second clock.

41. (New) A method of transmitting data across a high-speed serial bus, the method comprising:

in accordance with a first clock:

receiving a plurality of 8-bit bytes on an 802.3-compliant PHY;

deleting any null symbol appearing in any of the 8-bit bytes, wherein the frequency of null character deletion is used to control a phased locked loop associated with a second clock;

assembling a plurality of 10-bit symbols from the plurality of 8-bit bytes and from the contents of a register, wherein the register is adapted to store bits from an 8-bit byte that were not used in constructing a prior 10-bit symbol;

placing the assembled 10-bit symbols in a first FIFO;

in accordance with the second clock:

removing the 10-bit symbols from the first FIFO,

decoding each removed 10-bit symbol;

placing each decoded 10-bit symbol in a second FIFO;

in accordance with a third clock:

removing each decoded 10-bit symbol from the second FIFO; and

sending each decoded 10-bit symbol to an IEEE 1394-compliant PHY.

42. (New) A method of transmitting data across a high-speed serial bus, the method comprising:

receiving a first bit sequence on an 802.3-compliant PHY;

in accordance with a GMII RX clock:

if the first bit sequence does not contain a null symbol, then storing the first bit sequence;

else, if the first bit sequence contains a null symbol, then deleting the null symbol, wherein the frequency of null symbol deletion is used to control a phased locked loop associated with a second clock;

receiving a second bit sequence that does not contain a null symbol and storing the second bit sequence;

assembling a symbol based at least in part upon the first and second bit sequences; and

placing the symbol in a first FIFO data structure;

in accordance with a second clock:

removing the symbol from the first FIFO data structure;

decoding on the removed symbol; and

placing the decoded symbol in a second FIFO data structure;

in accordance with a third clock:

removing the decoded symbol from the second FIFO; and

sending the decoded symbol to an IEEE 1394-compliant PHY.